



# Battleship: El Juego

(Documentación)

# Contenido

Un Vistazo al Programa.....	3
<b>Funcionamiento de la Pantalla Principal:</b> .....	4
<b>Selección de Dificultad:</b> .....	5
<b>Nombres de los Jugadores:</b> .....	6
Pantalla de Selección de Coordenadas: .....	7
<b>Coordenada Inicial:</b> .....	8
<b>Escoger la Orientación del Barco:</b> .....	9
<b>Funcionamiento de la Orientación:</b> .....	10
<b>Diferencias entre los demás casos:</b> .....	10
<b>Completando el Tablero:</b> .....	11
Pantalla del Juego:.....	11
<b>Funcionamiento del juego:</b> .....	12
Pantalla de Resultados:.....	14
Listado de Variables Utilizadas (Con su Función):.....	16

## Un Vistazo al Programa

Al abrir el archivo de battleship.exe, se mostrará la pantalla principal del mismo con las siguientes opciones:

```
BATTLESHIP: THE GAME (v1.0)
CREADO POR: JANKA

| NOTA: ES NECESARIO ACTIVAR MAYUS CON EL FIN DE DETECTAR LAS OPCIONES CORRECTAMENTE |

A) INICIAR UNA NUEVA PARTIDA
B) REGLAS DEL JUEGO

X: _
```

Para efectos del programa, muchas de las opciones (como el desplazarse entre los menús) requieren que el usuario active la tecla mayus, para evitar recibir un mensaje de error.

En esta versión del minijuego, están las opciones de

- A) **Iniciar una nueva partida:** al entrar en esta opción, se procede a la pantalla de selección de dificultad.
- B) **Reglas del juego:** para quienes no estén familiarizados con el juego, esta opción les muestra a los usuarios un resumen de lo que consiste cada una de las secciones principales.

```
REGLAS GENERALES DEL JUEGO:

BIENVENIDOS A BATTLESHIP, EL OBJETIVO DE ESTE JUEGO ES EL DE ELIMINAR
LO MAS RAPIDO QUE PUEDAN TODOS LOS BARCOS DE SUS Oponentes.

O: SELECCION DE COORDENADAS:

A CADA JUGADOR LE CORRESPONDE UBICAR SUS CINCO BARCOS (DE DIFERENTES CLASES)
DENTRO DEL RANGO DE UN TABLERO 8X8, LOS BARCOS NO PUEDEN EXCEDER LOS
LIMITES O SOBREPONER OTROS PREVIAMENTE UBICADOS.

O: JUEGO EN CURSO:

UNA VEZ EN PARTIDA, POR TURNOS, CADA JUGADOR TENDRA LA OPORTUNIDAD DE
SELECCIONAR UNA COORDENADA ESPECIFICA, EL PROGRAMA LE INFORMARA SI
LOS JUGADORES HAN ATINADO A UNO DE LOS PUNTOS DEL BARCO, O SIMPLEMENTE
HAN COLOCADO EL PUNTO SIN EFECTO ALGUNO.

Presione una tecla para continuar . . .
```

## Funcionamiento de la Pantalla Principal:

Al momento de iniciar el programa, como primer paso, se rellenan cada una de las matrices que los jugadores van a estar utilizando (tanto las de coordenadas como las de ataque), en esta ocasión, se ha dado el uso de matrices tridimensionales de manera que, tendríamos dos matrices descritas como: `naves[z][f][c]` y `ataque[z][f][c]`.

*Leyenda:*

*Z = Esta variable representa el número de matrices de cada tipo, en este programa se utiliza en las matrices tanto de naves como de ataque ocupando el primer "[ ]", de la siguiente forma: naves [2][8][8].*

*F = Esta variable representa la cantidad de filas de las matrices de cada tipo, nuevamente utilizada en ambos casos, ocupando el segundo "[ ]". naves [2][8][8].*

*C = Esta variable representa la cantidad de columnas de las matrices de cada tipo, utilizada en ambos casos ocupando el tercer "[ ]". naves [2][8][8].*

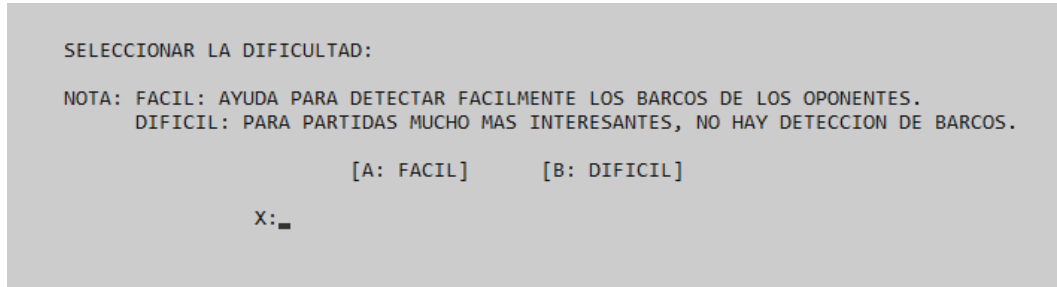
```
1066 void main(){
1067
1068     Tableros();
1069
1070     for(f=0;f<8;f++){
1071         for(c=0;c<8;c++){
1072             ataque[0][f][c]='~';
1073         }
1074     }
1075
1076     for(f=0;f<8;f++){
1077         for(c=0;c<8;c++){
1078             ataque[1][f][c]='~';
1079         }
1080     }
```

Para este programa, las matrices **inicialmente** se rellenan con el carácter "~", con el fin de dar la impresión de que esta en medio del mar. En la imagen de arriba se puede ver **el método aplicado** para las matrices de ataque.

Tras rellenar todas las matrices, entraría en un **ciclo while**, regulado por la variable `cmjug` o "Control Menú de Juego", y entonces hace el llamado al usuario para seleccionar alguna de las dos opciones. El uso del `switch` en este programa es abundante, esto debido a la facilidad para especificar que valores numéricos o caracteres se consideran como válidos. Por ejemplo, al entrar en la opción B, se realiza una limpieza de pantalla para mostrar las reglas y tras pulsar cualquier tecla regresa al menú principal.

## Selección de Dificultad:

Al haber seleccionado la opción A, entraríamos a la pantalla de selección de dificultad, con las siguientes categorías:



La dificultad a simple vista puede parecer como algo a considerar en caso de tener una partida contra una máquina en lugar de otra persona, pero en este caso consiste en una **pequeña adición**.

Es sencillo, al momento de estar en partida, el **modo fácil** ofrece a los jugadores la posibilidad de orientarse en cuanto a la posición de los barcos de sus oponentes, en otras palabras, al atinar en algún punto que contiene una parte de determinado barco, la coordenada seleccionada se marca con un "O", en lugar de una "X", la cual representaría un punto colocado sin efecto alguno, esto como una ayuda a los jugadores, opción la cual en el **modo difícil** no está disponible y que por defecto, todas las coordenadas independientemente de que contengan un punto de barco o no, se marcaran con una "X".

La variable que administra la dificultad sería "sdif" y puede tener un valor de "A" o de "B" según, la decisión de los jugadores, su aplicación se verá más adelante.

## Nombres de los Jugadores:

Se solicitan los nombres de ambos jugadores, estos están administrados por las variables jug1 y jug2. Al introducir los nombres, en caso tal se cometió un error, se pueden volver a modificar, aunque solo al comienzo.

```
void Jugadores(){  
    printf("\n\n\t\t\t\t\tINGRESE EL NOMBRE DEL JUGADOR 1\n");  
    printf("\t\t\t\t\tJ1: ");  
    gets(jug1);  
    fflush(stdin);  
    system("cls");  
  
    printf("\n\n\t\t\t\t\tINGRESE EL NOMBRE DEL JUGADOR 2\n");  
    printf("\t\t\t\t\tJ2: ");  
    gets(jug2);  
    fflush(stdin);  
    system("cls");  
}
```

NOMBRES DE LOS JUGADORES:

JUGADOR #1: JEAN CARLOS  
JUGADOR #2: RIVAS FOSKIN

\*\*\*CONFIRMAR NOMBRES O CAMBIARLOS\*\*\*

[1:CONFIRMAR]                      [0:CAMBIAR]

X:

### Pantalla de Selección de Coordenadas:

Tras la previa confirmación de los nombres, a continuación, cada uno de los jugadores por separado, tendrá la oportunidad de ingresar las coordenadas en las cuales orientar cada uno de los 5 barcos o naves que tiene a su disposición. Mientras esto ocurre, el otro jugador no puede observar la pantalla.

Esta sección se denota de esta forma:

```
JUGADOR #1: JEAN CARLOS
CONSIDERE LO SIGUIENTE:

|      00 = RESTANTE:2      |
|      000 = RESTANTE:2    |
|      0000 = RESTANTE:1   |
|                             |

  A  B  C  D  E  F  G  H
1 | ~ ~ ~ ~ ~ ~ ~ ~ |
2 | ~ ~ ~ ~ ~ ~ ~ ~ |
3 | ~ ~ ~ ~ ~ ~ ~ ~ |
4 | ~ ~ ~ ~ ~ ~ ~ ~ |
5 | ~ ~ ~ ~ ~ ~ ~ ~ |
6 | ~ ~ ~ ~ ~ ~ ~ ~ |
7 | ~ ~ ~ ~ ~ ~ ~ ~ |
8 | ~ ~ ~ ~ ~ ~ ~ ~ |

SELECCIONE EL TIPO DE BARCO QUE DESEA COLOCAR:

[2: 00] [3: 000] [4: 0000]

X: _
```

Se le hace saber al usuario, que tiene en sus manos, la opción de colocar un total de 5 barcos a lo largo del tablero 8x8 especificando **cuantos barcos** de cada tipo puede colocar, luego, muestra la matriz que inicialmente está llena de caracteres "~" pero que, con el tiempo, se van a ver reflejados los barcos colocados.

A continuación, un menú que pregunta al usuario, que tipo de barco desea colocar. Para efectos del **documento**, se mostrará el funcionamiento de uno de los tres tipos de barcos disponibles, con sus respectivas diferencias.

## Coordenada Inicial:

Al introducirnos en una de las opciones de tipo de barco, en primer lugar, se nos muestra el tipo de barco seleccionado, posteriormente el tablero, y seguido nos pregunta, donde queremos colocar el **punto inicial**, a partir de este punto, los usuarios podrán seleccionar la orientación de su preferencia, eso solo si, la coordenada especifica está **disponible**, en caso de no haber problema, el programa procede a mostrar la siguiente sección.

Se introduce la letra y posteriormente, el número de fila al cual se desea colocar el punto inicial, este proceso de selección de filas y columnas se administra junto con las funciones de Cambio\_Columna(); y Cambio\_Fila(); respectivamente.

BARCO 000:

	A	B	C	D	E	F	G	H
1	~	~	~	~	~	~	~	~
2	~	~	~	~	~	~	~	~
3	~	~	~	~	~	~	~	~
4	~	~	~	~	~	~	~	~
5	~	~	~	~	~	~	~	~
6	~	~	~	~	~	~	~	~
7	~	~	~	~	~	~	~	~
8	~	~	~	~	~	~	~	~

INDIQUE LAS COORDENADAS A LAS CUALES DESEA COLOCAR EL PUNTO INICIAL

NOTA: PRIMERO INDIQUE LA LETRA DE COLUMNA (EN MAYUS) Y  
POR CONSIGUIENTE EL NUMERO DE FILA |

Y:

```
void Cambio_Columna(){
    switch(y){
        case 'A': c=0; break;
        case 'B': c=1; break;
        case 'C': c=2; break;
        case 'D': c=3; break;
        case 'E': c=4; break;
        case 'F': c=5; break;
        case 'G': c=6; break;
        case 'H': c=7; break;
        default: printf("\n\t\t\t\tLETRA SELECCIONADA NO ESTA DENTRO DEL RANGO\n\t\t\t\t"); system("pause"); system("cls");
    }
}
void Cambio_Fila(){
    switch(x){
        case '1': f=0; break;
        case '2': f=1; break;
        case '3': f=2; break;
        case '4': f=3; break;
        case '5': f=4; break;
        case '6': f=5; break;
        case '7': f=6; break;
        case '8': f=7; break;
        default: printf("\n\t\t\t\tNUMERO SELECCIONADO NO ESTA DENTRO DEL RANGO\n\t\t\t\t"); system("pause"); system("cls");
    }
}
```



## Escoger la Orientación del Barco:

Tras escoger la coordenada inicial A3 (como ejemplo), se procede a preguntar por la orientación, el programa le ofrece **cuatro opciones** de orientación: arriba, abajo, izquierda y derecha. En caso tal todas las orientaciones obstaculizan con algunos de los puntos de los barcos, el usuario tiene la opción de volver a escoger las coordenadas, el hacer esto, hace que regrese a la pantalla anterior, volviendo el punto escogido de "O" a "~".

```
UBICACION DEL BARCO: 000

  A B C D E F G H
1 | ~ ~ ~ ~ ~ ~ ~ ~ |
2 | ~ ~ ~ ~ ~ ~ ~ ~ |
3 | O ~ ~ ~ ~ ~ ~ ~ |
4 | ~ ~ ~ ~ ~ ~ ~ ~ |
5 | ~ ~ ~ ~ ~ ~ ~ ~ |
6 | ~ ~ ~ ~ ~ ~ ~ ~ |
7 | ~ ~ ~ ~ ~ ~ ~ ~ |
8 | ~ ~ ~ ~ ~ ~ ~ ~ |

ELIJA LA ORIENTACION DEL BARCO

NOTA: SI TODAS LAS ORIENTACIONES OBSTACULIZAN CON ALGUNOS DE
LOS BARCOS, PUEDE VOLVER A ESCOGER LAS COORDENADAS

[1: ARRIBA ] [2: ABAJO]
[3: IZQUIERDA] [4: DERECHA]

[R:CAMBIAR COORDENADAS]

X: _
```

Al escoger el caso cuatro (como demostración), el programa regresa a la pantalla de selección de tipo de barco, mostrando la matriz junto con los cambios realizados tanto en valores restantes como en el propio tablero.

```
JUGADOR #1: JEAN CARLOS
CONSIDERE LO SIGUIENTE:

|         OO = RESTANTE:2         |
|         OOO = RESTANTE:1        |
|         OOOO = RESTANTE:1       |

  A B C D E F G H
1 | ~ ~ ~ ~ ~ ~ ~ ~ |
2 | ~ ~ ~ ~ ~ ~ ~ ~ |
3 | O O O ~ ~ ~ ~ ~ |
4 | ~ ~ ~ ~ ~ ~ ~ ~ |
5 | ~ ~ ~ ~ ~ ~ ~ ~ |
6 | ~ ~ ~ ~ ~ ~ ~ ~ |
7 | ~ ~ ~ ~ ~ ~ ~ ~ |
8 | ~ ~ ~ ~ ~ ~ ~ ~ |

SELECCIONE EL TIPO DE BARCO QUE DESEA COLOCAR:

[2: 00] [3: 000] [4: 0000]

X: _
```

## Funcionamiento de la Orientación:

Mientras se este escogiendo la orientación, el programa debe comprobar si los espacios en lo que escribirá las "O" restantes son válidos y también si intenta sobreponer algún espacio ocupado por otro barco, este seria un ejemplo de como funcionaria con el caso de los barcos de tamaño OOOO:

```
case '4': m=c+1; n=c+2; o=c+3;
if(m!=8 && n!=8 && o!=8){
    if((naves[z][f][c+1]!='O') && (naves[z][f][c+2]!='O') && (naves[z][f][c+3]!='O') ){
        naves[z][f][c+1]='O';
        naves[z][f][c+2]='O';
        naves[z][f][c+3]='O';
        cont++; contcolumn++;
        m=0; n=0; o=0;
    }
    else{
        printf("\n\t\tLA ORIENTACION DE SU NAVE ESTA SOBREPONIENDO OTRA NAVE,SELECCIONE OTRA\n\t\t\t");
        system("pause"); system("cls");
    }
}
else{
    printf("\n\t\tLA ORIENTACION DEL BARCO SOBREPASA LOS LIMITES DEL TABLERO\n\t\t\t");
    system("pause"); system("cls");
}
break;
```

Tenemos, además, otras variables m, n y o, las cuales (dependiendo del tipo de barco seleccionado, estarán sumando o restando valores desde la posición escogida por el usuario, esto más que nada para comprobar si dichas posiciones son existentes, es decir, se encuentren dentro del rango del tablero 8x8 que en el programa realmente seria de valores entre 0-7 para ambos (fila, columna).

El caso cuatro es de DERECHA, con lo cual se estaría sumando desde la columna del punto inicial, Ejm: Coordenada Inicial: A5 = (0,4), entonces al seleccionar el case '4' seria m=0+1, n=0+2 y o= 0+3. De ser IZQUIERDA, entonces se restaría desde la columna inicial y con las de ARRIBA Y ABAJO, igual, solamente que cambiaría en lugar de columnas, comprobar con las filas correspondientes.

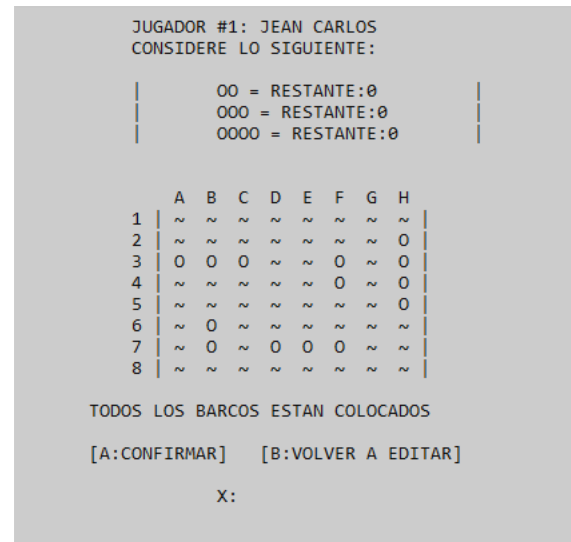
## Diferencias entre los demás casos:

La diferencia que hay entre un tipo de barco u otro, es, aparte de la selección de más de uno del mismo, es el hecho de utilizar las variables m, n y o. En el caso dos, solo debe comprobar **una posición**, en el caso tres **dos posiciones** y en el caso cuatro, **tres**, a cada posición se le asigna una de estas variables.

En cuanto al resto, es prácticamente **el mismo código**, partes como: la selección del punto inicial, si desea cambiarlo, eso sí, ajustado de modo que únicamente realice los cambios en tipo de barco concreto.

## Completando el Tablero:

Una vez colocados todos los barcos, se le mostrara al jugador en pantalla, si desea volver a cambiarlos o si directamente desea confirmar. Decidir cambiarlos borrara todas las configuraciones anteriores con el fin de poder ingresar nuevamente a los tres casos, por el contrario, al confirmar tu selección de coordenadas, hará que se guarde y entonces sería el turno **del segundo jugador** de introducir su configuración de coordenadas siguiendo los mismos pasos.



## Pantalla del Juego:

Una vez que ambos jugadores hayan introducido sus respectivos barcos, se procede al juego, el procedimiento dentro de la función es sencillo, se ingresa a un while el cual administra el turno del jugador 1 (tjug1), una vez cumplidos los requisitos, accede a otro con el turno del jugador 2 (tjug2). Para efectos del programa, el primer jugador en obtener 14 puntos es el vencedor (siendo cada punto, por coordenada de barco atinada).

## Funcionamiento del juego:

```

                                MODO DE JUEGO: FACIL

                                JUGADOR #1: JEAN CARLOS

                                PUNTOS OBTENIDOS: 0

                                A B C D E F G H
                                1 | ~ ~ ~ ~ ~ ~ ~ ~ |
                                2 | ~ ~ ~ ~ ~ ~ ~ ~ |
                                3 | ~ ~ ~ ~ ~ ~ ~ ~ |
                                4 | ~ ~ ~ ~ ~ ~ ~ ~ |
                                5 | ~ ~ ~ ~ ~ ~ ~ ~ |
                                6 | ~ ~ ~ ~ ~ ~ ~ ~ |
                                7 | ~ ~ ~ ~ ~ ~ ~ ~ |
                                8 | ~ ~ ~ ~ ~ ~ ~ ~ |

                                SELECCIONE UNA COORDENADA

                                Y:A
                                X:1

                                | PUNTO COLOCADO |

                                Presione una tecla para continuar . . .

```

Se muestra en pantalla, el nombre junto con un contador de los puntos totales y posteriormente el tablero de ataque del jugador que tiene el primer turno. Los jugadores deben seleccionar una coordenada valida, en caso tal se seleccione una coordenada llena, el programa le volverá a preguntar por otro valor, con el fin de no perder el turno, una vez seleccionada pueden ocurrir dos cosas:

- a) El jugador le atina a uno de los puntos del barco enemigo y se le sumaria +1 a su contador.
- b) Se colocaría el punto como tal, sin ningún cambio, y pasaría el turno al siguiente jugador.

El proceso se da por turnos y como se había dicho anteriormente, la partida terminara únicamente cuando alguno de los dos jugadores consiga los 14 puntos, todo esto se administra mediante el siguiente código:

```
void Juego(){  
    while(game==0){  
  
        while(tjug1==0){  
  
            if(sdif=='A'){ printf("\n\t\t\t\t\t MODO DE JUEGO: FACIL\n\n"); }  
            else if(sdif=='B'){ printf("\n\t\t\t\t\t MODO DE JUEGO: DIFICIL\n\n"); }  
  
            printf("\t\t\t\tJUGADOR #1: %s\n", jug1);  
            printf("\t\t\t\t PUNTOS OBTENIDOS: %i\n", pjug1);  
  
            Display_Tb_Ataque_J1()  
  
            printf("\n\t\t\t\t\t SELECCIONE UNA COORDENADA\n\n");  
            printf("\t\t\t\t\t Y:");  
            scanf("%c",&y);  
            fflush(stdin);  
            Cambio_Columna();  
  
            if(c>=0 && c<=7){  
                printf("\t\t\t\t\t X:");  
                scanf("%c",&x);  
                fflush(stdin);  
                Cambio_Fila();
```

```
if(f>=0 && f<=7){
    if(ataque[0][f][c]=='~'){
        ataque[0][f][c]='X';
        if(naves[1][f][c]=='O'){
            ataque[0][f][c]='O';
            if(sdif=='B'){ ataque[0][f][c]='X'; }
            printf("\n\t\t\t\t\t\t\t\t\t\t ATINADO +1 |\n\n\t\t\t\t\t\t\t\t");
            tjugl++;
            pjugl++;
            f=-1;
            c=-1;
            system("pause");

            if(pjugl==14){
                system("cls");
                return;
            }
            system("cls");
        }
    } else if(naves[1][f][c]=='~'){
        printf("\n\t\t\t\t\t\t\t\t\t\t PUNTO COLOCADO |\n\n\t\t\t\t\t\t\t\t\t\t");
        tjugl++;
        f=-1;
        c=-1;
        system("pause");
        system("cls");
    }
}

else{
    printf("\n\t\t\t\t\t\t\t\t\t\tTESTE UBICACION YA ESTA MARCADA, SELECCIONE UNA DIFERENTE\n\t\t\t\t\t\t\t\t\t\t");
    system("pause");
    system("cls");
}
```

Como se puede apreciar, el uso de la variable sdif, influye en los aspectos del cambio de una coordenada correcta pasando de un 'O' a 'X' y en la propia etiqueta del modo de juego.

## Pantalla de Resultados:

Cuando alguno de los dos jugadores haya obtenido los 14 puntos, la función entraría en un if específico junto con el valor return, al realizar el return a la función principal, lo siguiente en el código sería mostrar la pantalla de los resultados mediante Resultados();

EL GANADOR DE ESTA PARTIDA ES JEAN CARLOS ,FELICIDADES!

A CONTINUACION SE MUESTRAN LAS COORDENADAS DE AMBOS JUGADORES  
PARA COMPROBAR LAS UBICACIONES DE LOS BARCOS JUNTO CON LAS DE  
ATAQUE DE CADA JUGADOR

Presione una tecla para continuar . . .

UBICACION DE BARCOS DEL JUGADOR #1

	A	B	C	D	E	F	G	H
1	O	O	~	~	~	O	O	O
2	~	~	~	~	~	~	~	~
3	~	~	~	~	~	~	~	~
4	~	~	~	~	~	~	~	O
5	~	~	~	~	~	~	~	O
6	~	~	~	~	~	~	~	O
7	O	~	~	~	~	~	~	O
8	O	~	~	~	~	O	O	O

UBICACION DE COORDENADAS DE ATAQUE DEL JUGADOR #2

	A	B	C	D	E	F	G	H
1	~	~	X	X	X	O	O	O
2	X	~	X	X	~	X	X	~
3	~	X	X	X	~	~	X	~
4	X	X	X	~	~	X	X	O
5	X	X	X	~	X	X	X	~
6	~	X	~	~	X	~	X	O
7	O	X	X	X	~	~	X	O
8	~	X	~	~	~	~	~	O

UBICACION DE BARCOS DEL JUGADOR #2

	A	B	C	D	E	F	G	H
1	~	~	~	~	~	~	~	~
2	~	~	~	~	~	~	~	~
3	~	~	~	~	~	~	~	~
4	~	~	~	O	O	O	O	O
5	~	~	~	O	O	O	~	~
6	~	~	~	O	~	~	~	~
7	~	~	~	~	~	~	O	~
8	~	~	O	O	O	~	O	~

UBICACION DE COORDENADAS DE ATAQUE DEL JUGADOR #1

	A	B	C	D	E	F	G	H
1	X	~	~	~	~	~	~	~
2	X	~	X	~	~	~	~	~

Luego de haber pasado la pantalla de resultados, el programa les preguntara a los jugadores si desean comenzar otra partida o si ya desean terminar de jugar. El seleccionar la opción de volver a jugar, reiniciara todas las variables a sus valores iniciales, y repetirá el proceso como si se abriese de nuevo el archivo .exe.

EN CASO DE QUERER JUGAR OTRA PARTIDA, INGRESE LA OPCION CORRESPONDIENTE

[A: INICIAR OTRA PARTIDA]

[B: SALIR DEL JUEGO]

 $X:$ 

Eso sería todo en cuanto a esta versión del juego. Antes de finalizar, se presentará un listado de todas las variables utilizadas, junto con su función dentro del programa.

## Listado de Variables Utilizadas (Con su Función):

---

- **i:** Variable que sustituye la de las filas (f), para no modificar su valor mientras se realiza la selección de coordenadas, usada mayormente en las funciones Display de ambos tipos de tablero.
- **j:** Variable que sustituye la de las columnas (c), para no modificar su valor mientras se realiza la selección de coordenadas, usada mayormente en las funciones Display de ambos tipos de tableros.
- **f:** Variable que representa las filas de las matrices.
- **c:** Variable que representa las columnas de las matrices.
- **z:** Variable que representa la cantidad de matrices de cada tipo (ataque y naves).
- **m:** usada para sumar o restar valores desde columna o fila específica (comprobación si sigue dentro del tablero 8x8 durante la selección de orientación), se ve aplicada en todos los casos de barcos.
- **n:** usada para sumar o restar valores desde columna o fila específica (comprobación si sigue dentro del tablero 8x8 durante la selección de orientación), se ve aplicada en el caso de barco OOO.
- **o:** usada para sumar o restar valores desde columna o fila específica (comprobación si sigue dentro del tablero 8x8 durante la selección de orientación), se ve aplicada en el caso de barco OOOO.
- **confb2:** utilizada para asegurar que solo se tomen dos barcos del tipo OO.
- **confb3:** utilizada para asegurar que solo se tomen dos barcos del tipo OOO.
- **valb2, valb3, valb4:** variables que regulan que se esté dentro de la sección de un tipo de barco específico, hasta confirmar la selección de coordenadas y orientación correspondientes, igualmente que las de confb2 y confb3, aseguran que no se tomen más que solo los valores restantes de barcos dados.
- **infcolumn:** muestra la información del tipo de barco, el Display de la matriz del jugador que esta en proceso de selección de coordenadas y la columna escogida en caso tal se haya elegido un valor de fila incorrecto.
- **game:** esta variable se asegura de que se repitan los turnos de los 2 jugadores, hasta que uno haya conseguido los puntos requeridos para ganar la partida.
- **selfila:** variable que repite el proceso de selección de fila hasta conseguir un valor correcto y dentro del rango.
- **cont:** variable que repite el proceso de selección de orientación hasta conseguir un valor valido y dentro del rango.



- **nomval:** variable que repite el proceso de escritura de los nombres de los jugadores hasta que se confirmen los mismos.
- **bart2, bart3, bart4:** contadores que informan al jugador de cuantos barcos de cada tipo tiene restantes.
- **totbarc:** variable que asegura que todos los jugadores seleccionen coordenadas para los cinco barcos definidos. Se resetea en caso tal se desea volver a editar el tablero (al tenerlo listo) y luego de que el primer jugador confirme para permitir al siguiente jugador hacer su respectiva selección.
- **tjug1, tjug2:** regulan que ambos jugadores tengan su turno de ataque durante la partida en curso. Reseteadas al final del turno del jugador 2 para volver a repetir turnos gracias a la variable game.
- **pjug1, pjug2:** llevan el puntaje obtenido por los jugadores durante la partida.
- **salid:** variable que repite la pregunta de "si desea volver a jugar otra partida" hasta que se haya seleccionado una de las opciones correspondientes.
- **cmjug:** variable que repite la pantalla de título, hasta que el usuario haga selección de una dificultad para proceder con el ingreso de nombres.
- **jug1, jug2:** variables que reciben los nombres de los jugadores y los muestra durante la selección de coordenadas (para saber de quien es el turno) y en la partida, junto con los puntajes de ambos.
- **x:** recibe y controla que el valor ingresado de fila o sea el correcto, de ser así hace el cambio para adaptarse al tamaño de la matriz.
- **y:** recibe y controla que el valor ingresado de columna sea el correcto, de ser así hace el cambio para adaptarse al tamaño de la matriz.
- **orint:** recibe el valor de orientación seleccionada por el usuario, ARRIBA, ABAJO, DERECHA, IZQUIERDA O de cambio en caso tal se complique.
- **opc1, opc:** variables con usos diversos, reciben valores numéricos o letras dependiendo de las opciones seleccionadas por los jugadores. Para confirmar, cambiar, regresar entre los menús, etc.
- **sdif:** control de la dificultad escogida por los jugadores.
- **naves & ataque:** matrices tridimensionales utilizadas durante el programa (tipo char), inicializadas con el carácter "~" y con variaciones de "O" y "X".